Application of Linux Audio in Hearing Aid Research

Giso Grimm[1] Tobias Herzke[2] Volker Hohmann[2]

[1]Universität Oldenburg, Oldenburg, Germany

[2]HörTech gGmbH, Oldenburg, Germany

Linux Audio Conference, April 17th, 2009.

Hearing Aids

Analogue hearing aids

- Limited capabilities:
- Frequency shaping and amplification
- Dynamic compression
- Static notch filters for feedback cancellation

Hearing Aids

Analogue hearing aids

- Limited capabilities:
- Frequency shaping and amplification
- Dynamic compression
- Static notch filters for feedback cancellation

Digital hearing aids

- First hearing aid with Digital Signal Processing in 1996
- New algorithms with no analogue counterpart
- Wireless binaural link between ears
  - Parameter exchange
  - Coming soon: Low-delay low-power audio transmission

Limitations of hearing aids

Device

- Small battery capacity
- Low processing power

Acoustics

- Acoustic feedback
- Limited audio bandwidth (by tube between receiver and ear)

End user

- High expenses
- Unfulfilled expectations

Algorithms scalable to individual hearing loss

- Dynamic compression
- Frequency shaping
- Frequency compression

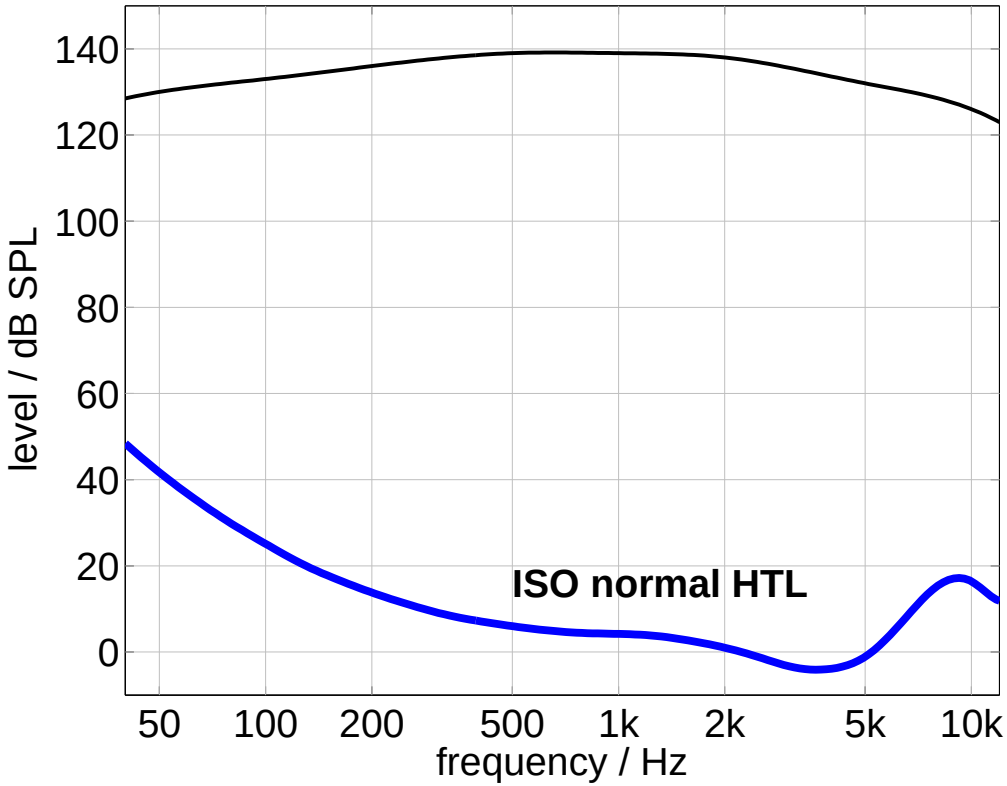
Signal enhancement

- Directional microphones
- Noise reduction

Artifact reduction

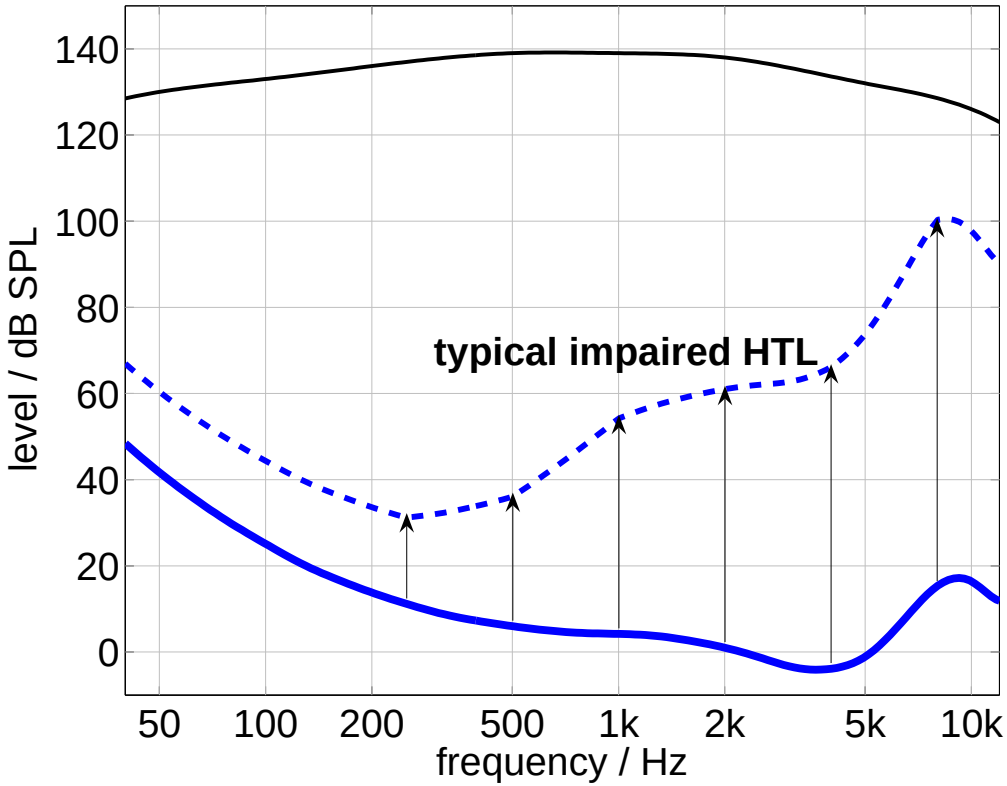
- Feedback cancellation
- Future: Non-linearity compensation, ...

Why dynamic compression?



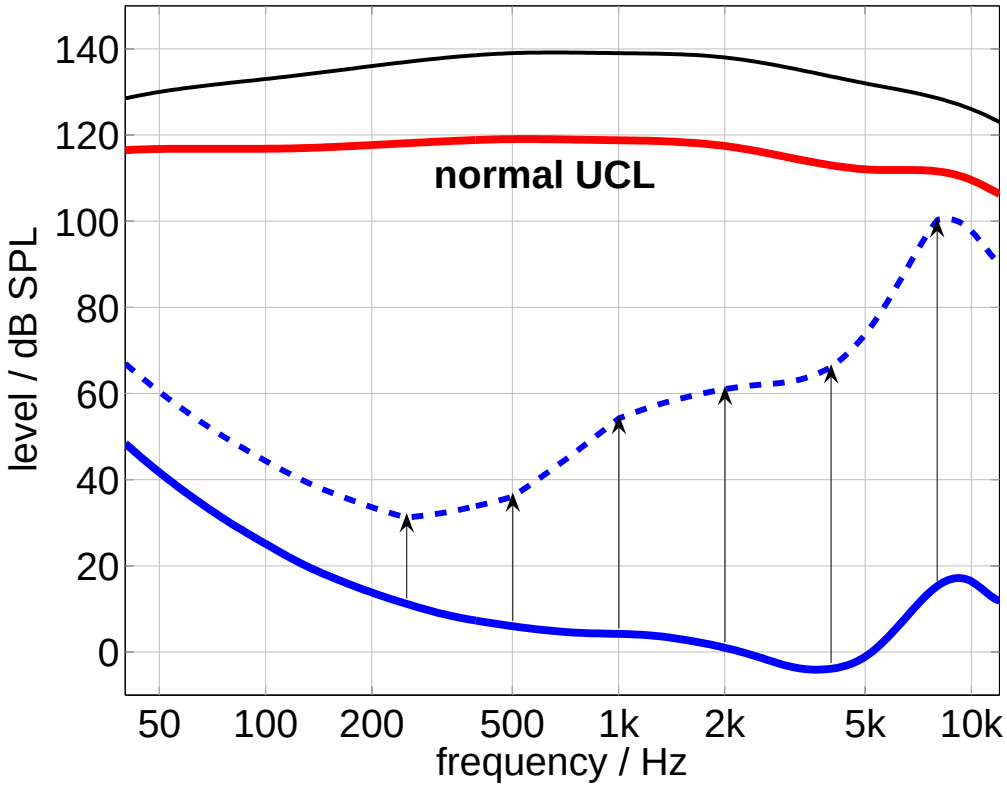
- Normal hearing threshold

Why dynamic compression?



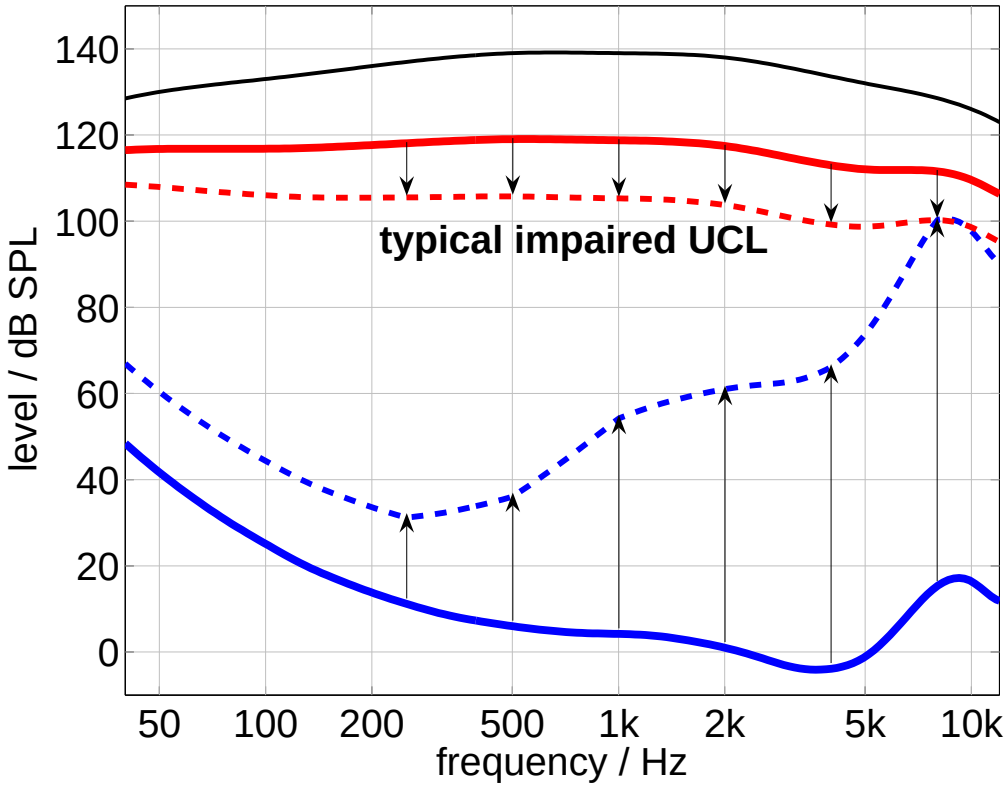
- Hearing impaired threshold
- Amplification and frequency shaping

Why dynamic compression?



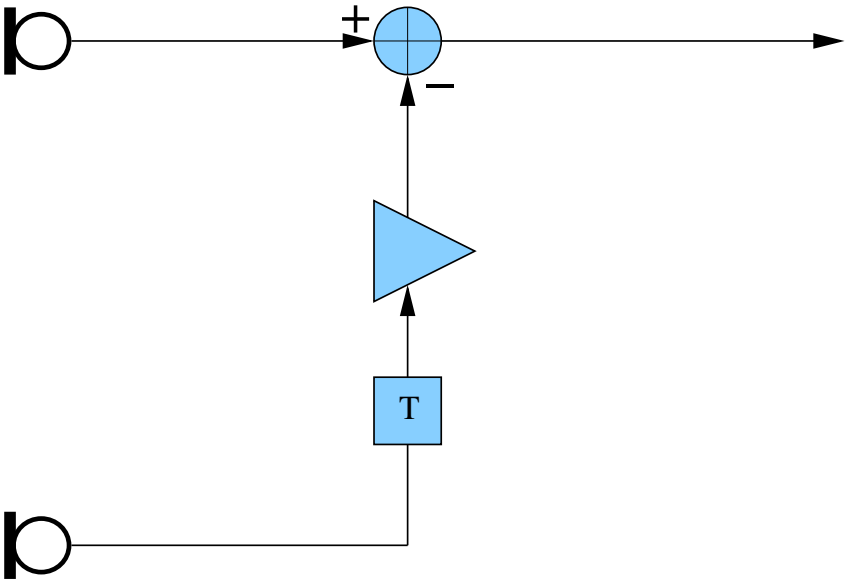
- Normal uncomfortable level

Why dynamic compression?



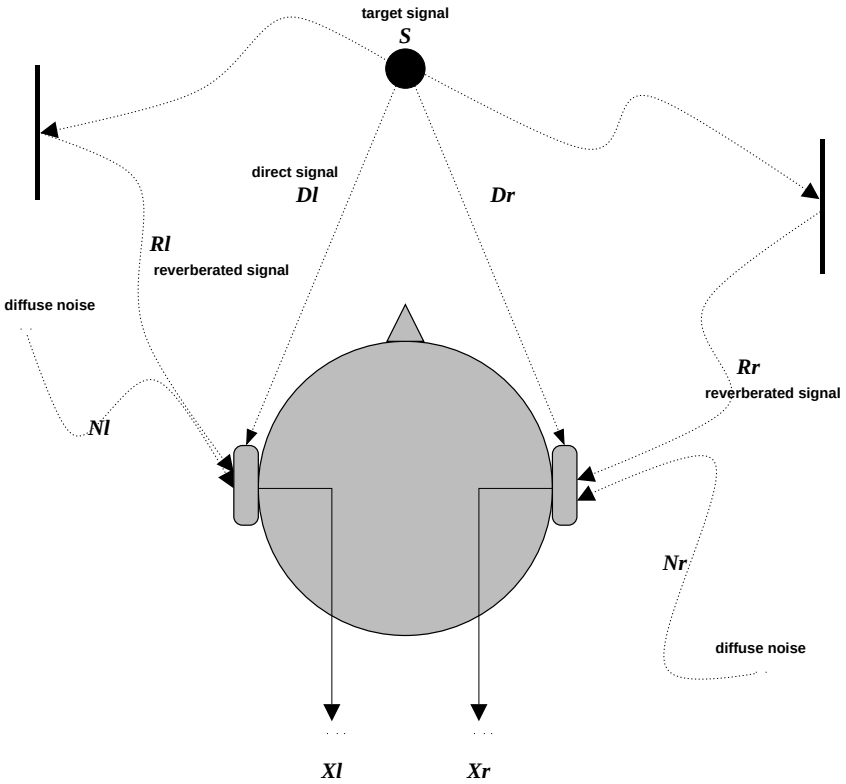
- Hearing impaired uncomfortable level
- Dynamic compression

Directional microphones: SNR improvement



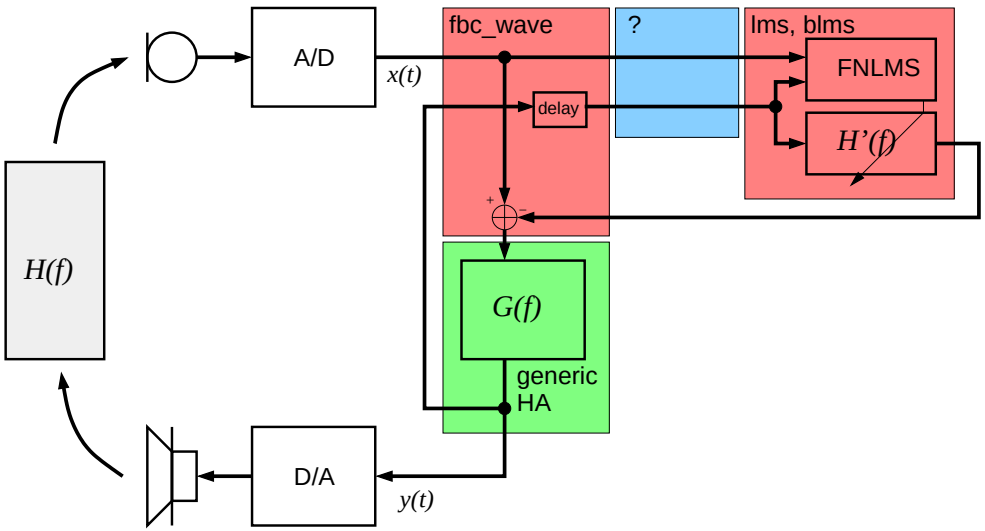
- Delay-and-sum: up to 3 dB SNR-improvement for on-axis sounds in diffuse noise
- Many more beamformers and direction-of-arrival estimators

Noise reduction and de-reverberation



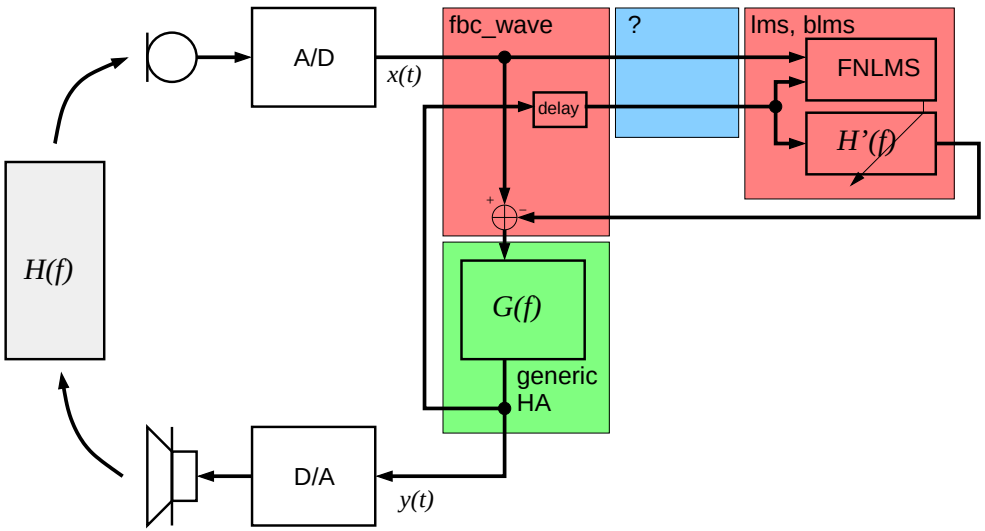
- Single channel noise reduction
- Binaural noise reduction, de-reverberation
- Binaural coherence estimation: Interaural phase difference statistics

Artifact reduction: Feedback cancellation



- Adaptive feedback cancellation
- Estimation of feedback signal
- Subtraction from input

Artifact reduction: Feedback cancellation



- Adaptive feedback cancellation
- Estimation of feedback signal
- Subtraction from input
- Other solutions: Frequency shifting, phase modulation, binaural coherence

Evaluation of Hearing Aids

Expected benefit of hearing aids

- Improvement of speech intelligibility
- Reduction of listening effort in adverse listening conditions
- Increase of listening comfort

Evaluation of Hearing Aids

Expected benefit of hearing aids

- Improvement of speech intelligibility
- Reduction of listening effort in adverse listening conditions
- Increase of listening comfort

Evaluation is required!

Evaluation methods

'Objective' methods

- SNR improvement (e.g., shadow filtering)
- Speech intelligibility index (f-weighted SNR)
- Quality prediction (usually similarity measures)

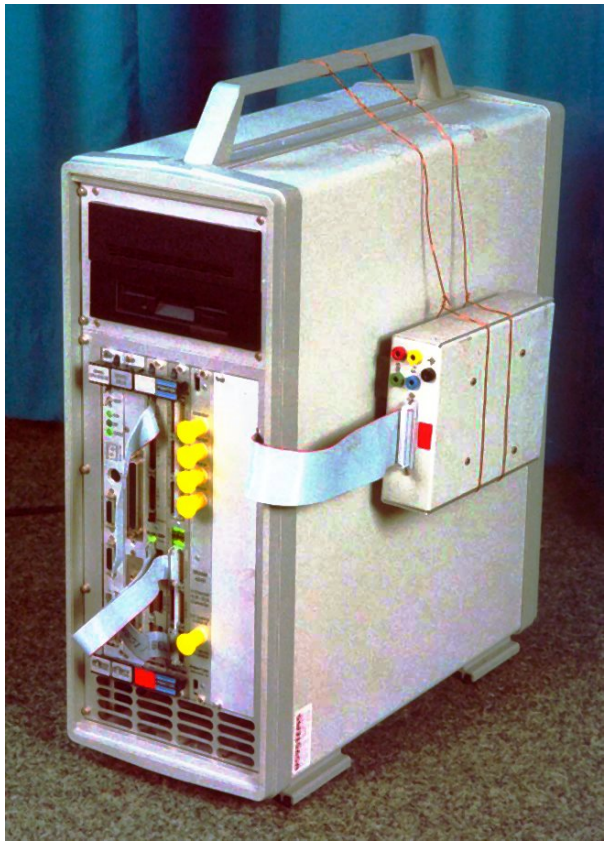
Subjective methods

- Speech recognition threshold in quiet and noise
- Quality rating, paired comparison ...

Test conditions

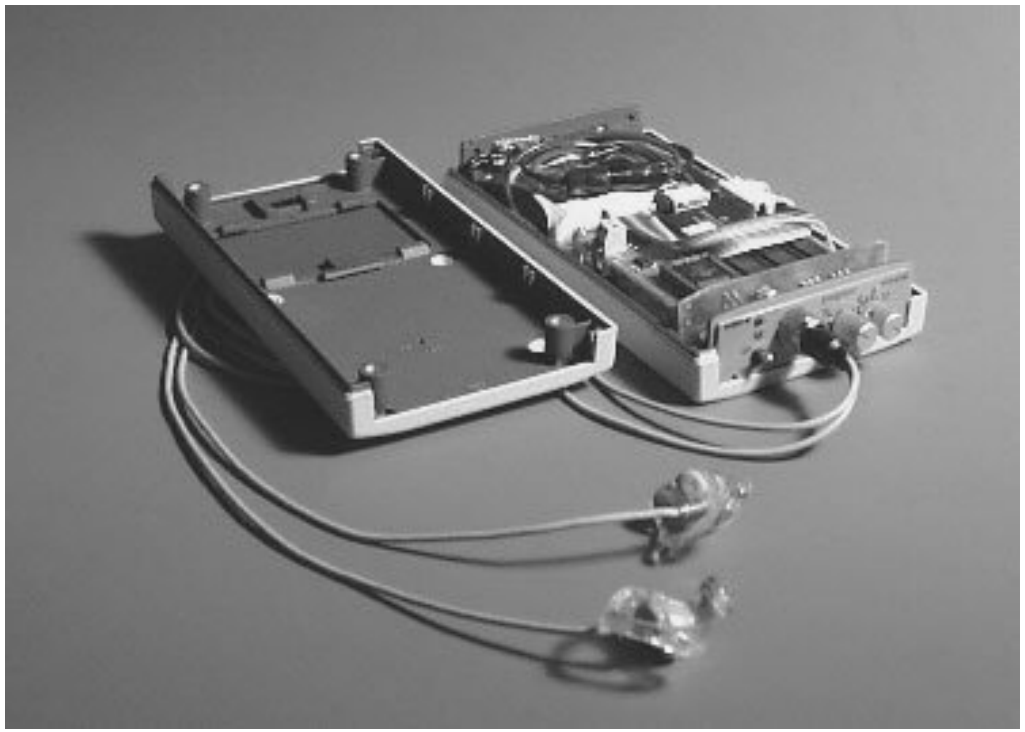
- 'static' situations:
  no spatial influecnce, pre-processed or real-time procesing
- dynamic conditions:
  head movements or moving sources, real-time processing
- realistic devices, acoustic feedback:
  low-delay real-time processing
- Real-life conditions:
  wide dynamic range, portable low-delay real-time processing

Real-time systems



1995: PC with DSP board, assembler programming

Real-time systems



2000: Portable DSP board, assembler programming
(DASi, U. Rass 2001)

Real-time systems

2008: Portable PC, C++ (or Matlab) programming

A Linux Hearing Aid

A Linux Hearing Aid

for research, development and field testing

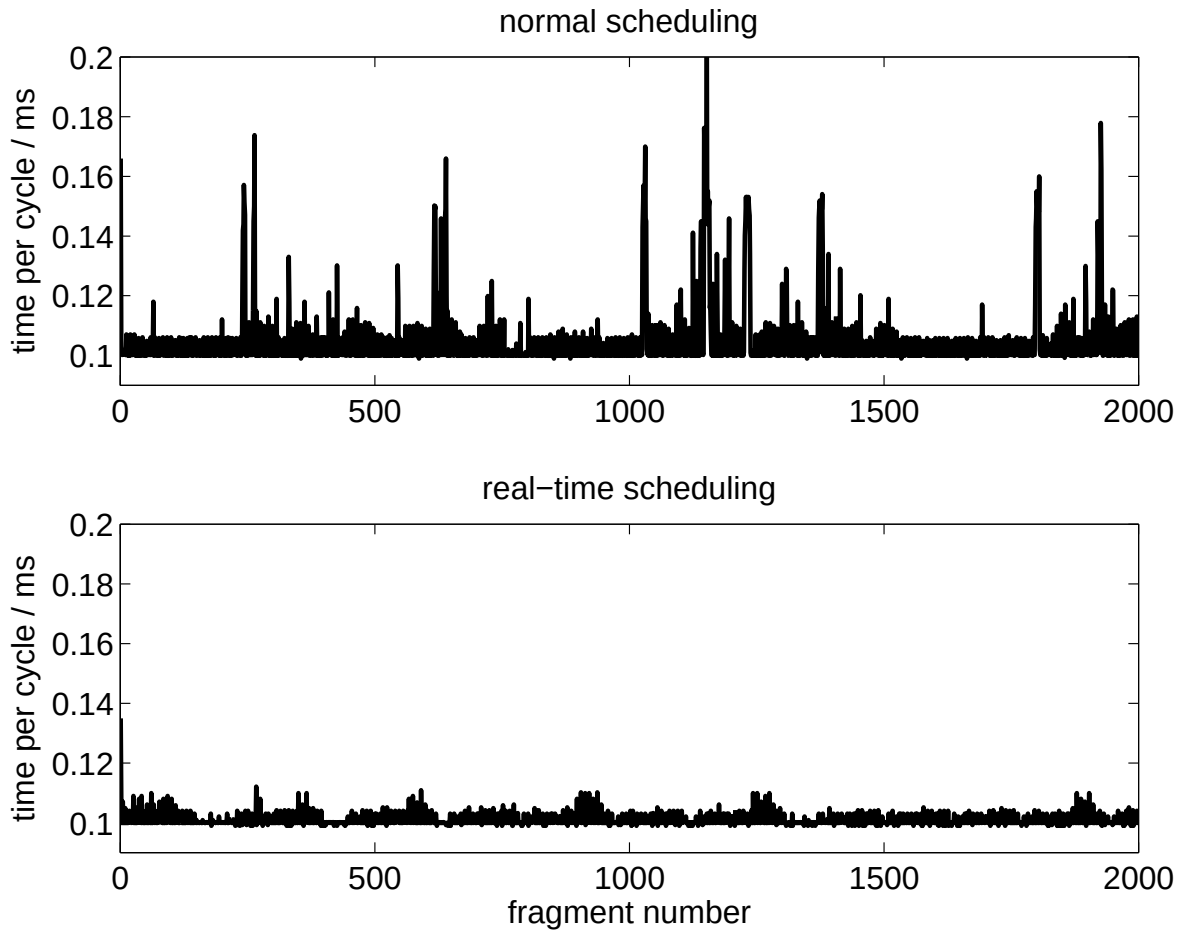
The 'Master Hearing Aid' software

- Platform for hearing aid algorithm development and evaluation
- Many audio backends: JACK, ALSA (Linux), ASIO (Windows), file, network, Matlab
- Extremely modular structure
- Hearing-aid (and hearing) related processing blocks
- Commercial product (closed source) by HoerTech gGmbH, Oldenburg
- Used in research projects and hearing aid industry

Licensing in competitive research

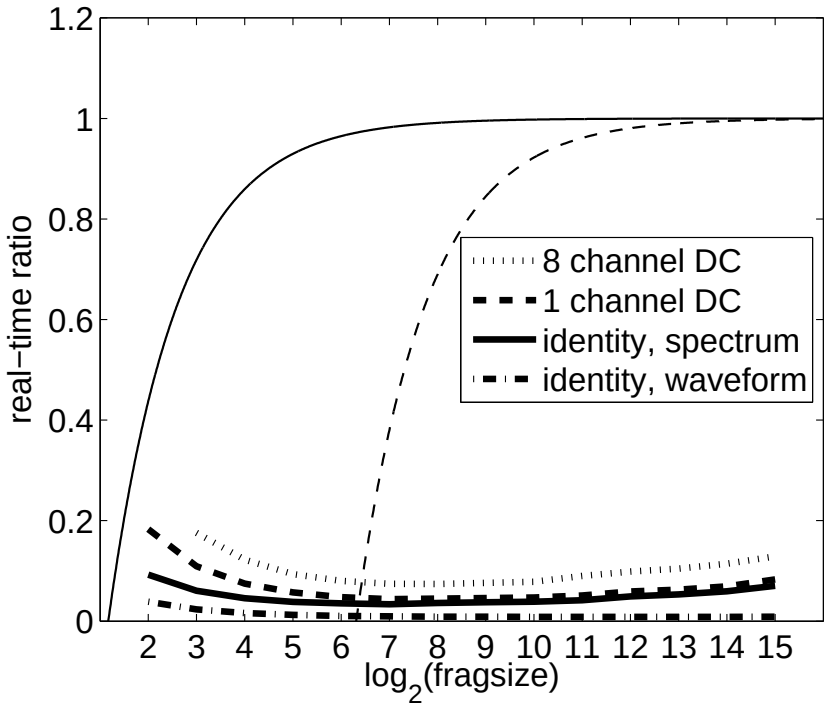
- Industrial/competitive context often requires closed-source development
- Industry decision makers fear the open source licensing of Linux
- It is possible to develop closed software on the Linux platform without infringing any licenses!
- Careful consideration of what components to use
- Knowledge of the relevant licenses, and compliance with their terms.
- Use of open source software in industrial context benefitial for both sides

Comparison of RT kernel versus non-RT kernel



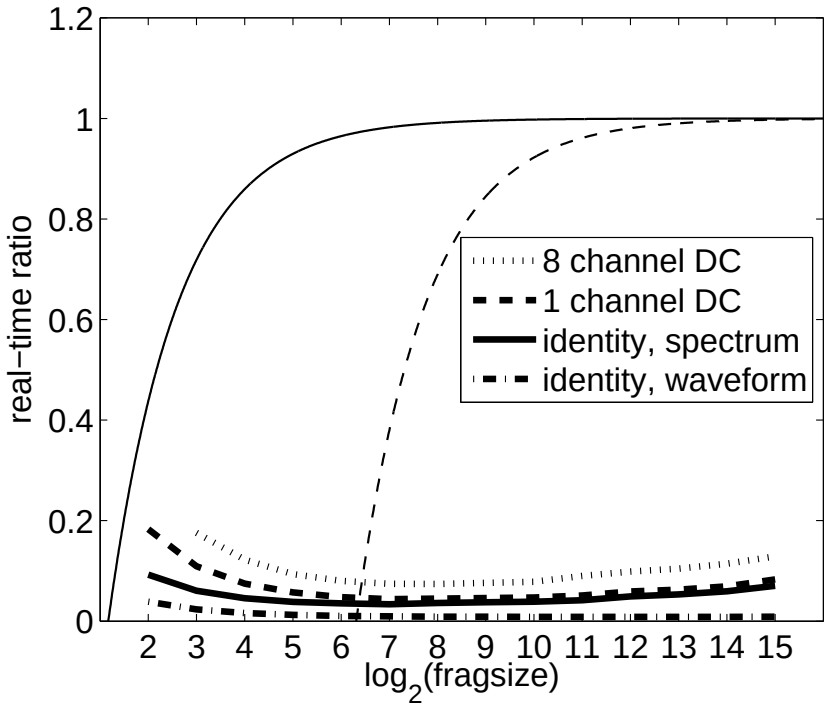
Kernel version 2.6.X (2.6.23?), from Grimm et al. (2006)

RT performance and low-delay

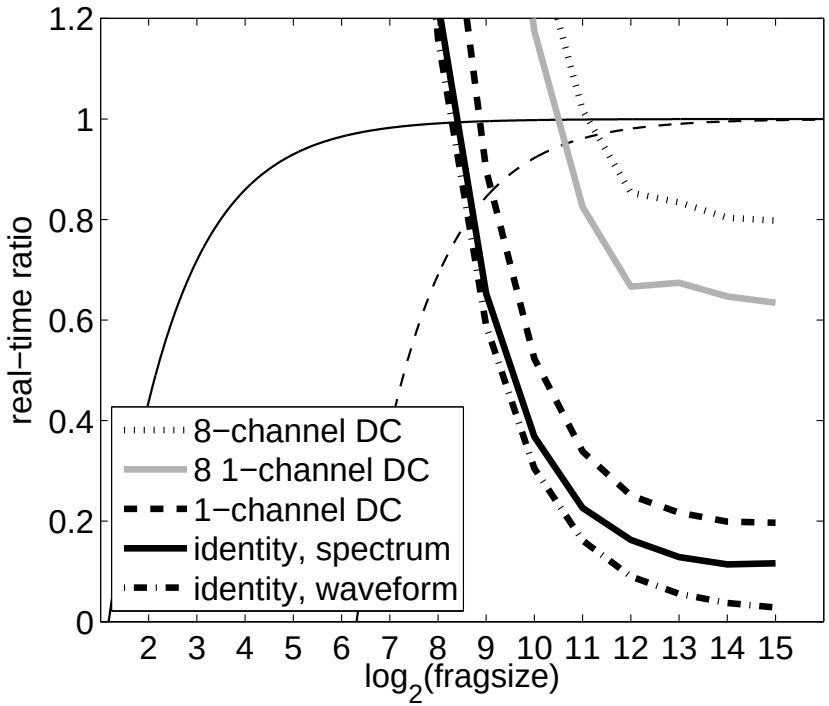


Algorithm implementation in C/C++

RT performance and low-delay



Algorithm implementation in C/C++

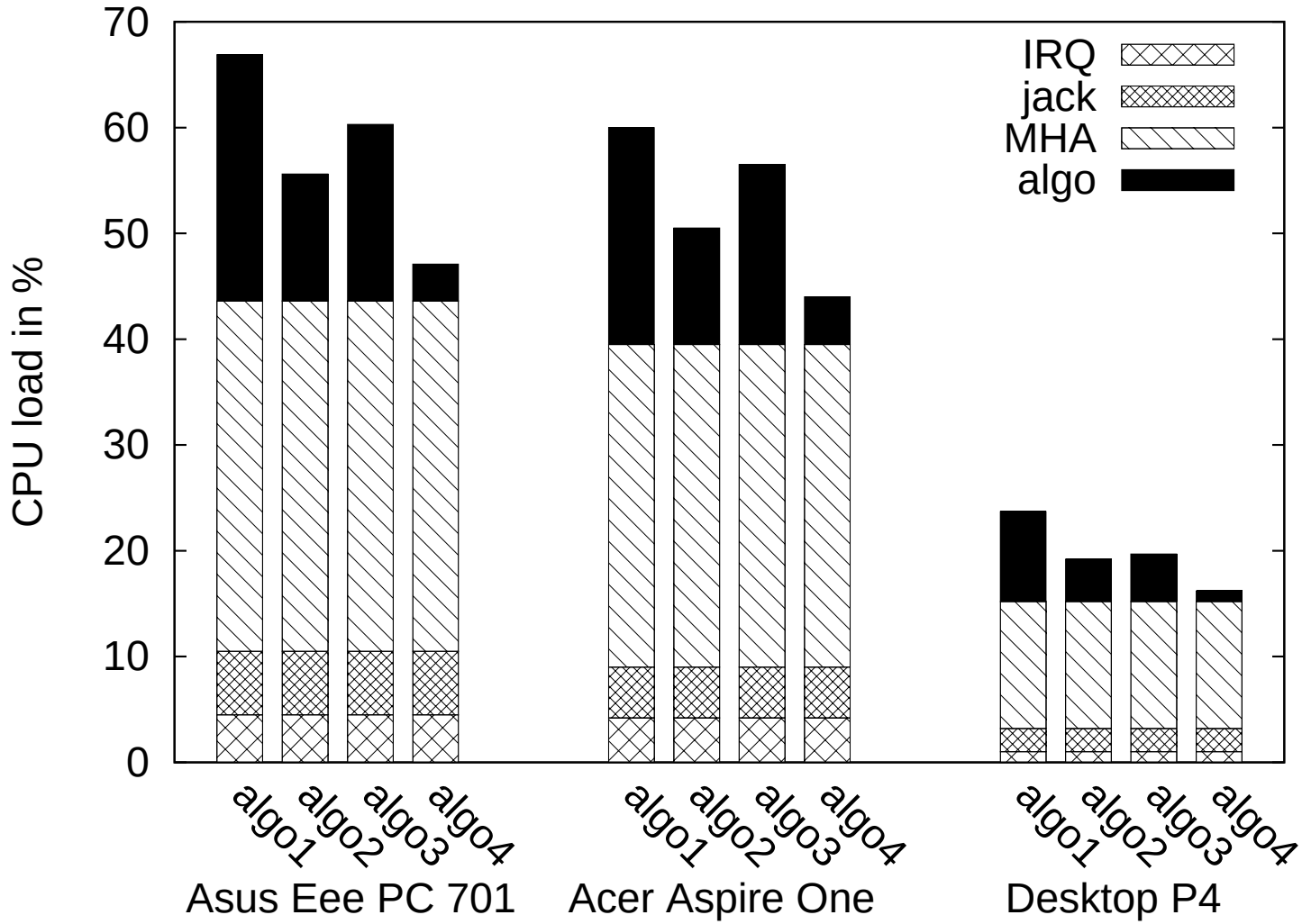


Algorithm implementation in Matlab

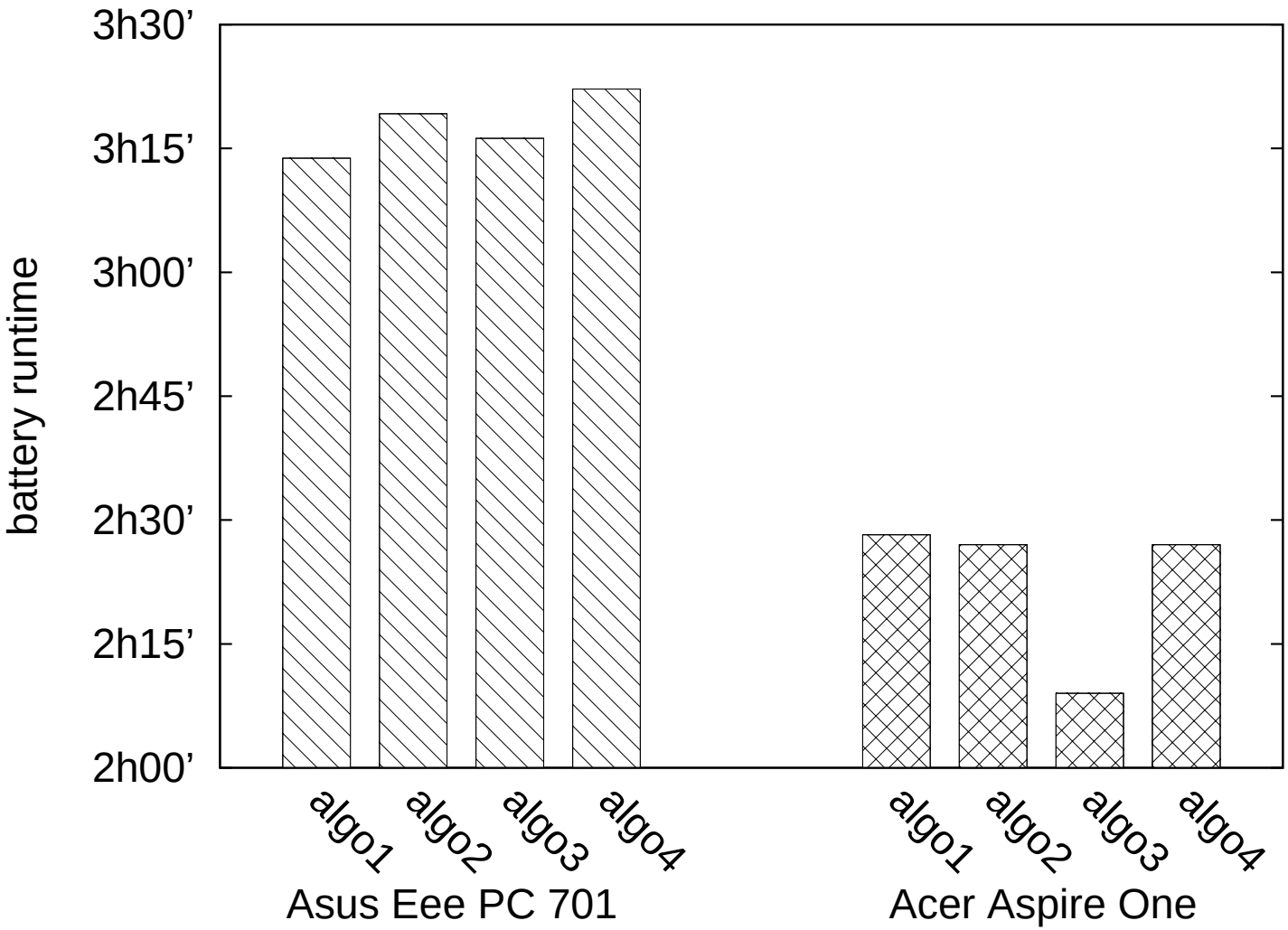
CPU and batttery performance

- Two portable systems
- Asus Eee PC 701:
  Intel Celeron @ 630 MHz
- Acer Aspire one:
  Intel Atom @ 1.6 GHz
- Reference Desktop system (Intel P4 @ 3GHz)

Comparison of CPU performance



Comparison of batteries



Delay Constraints

Three sources of delay in RT processing

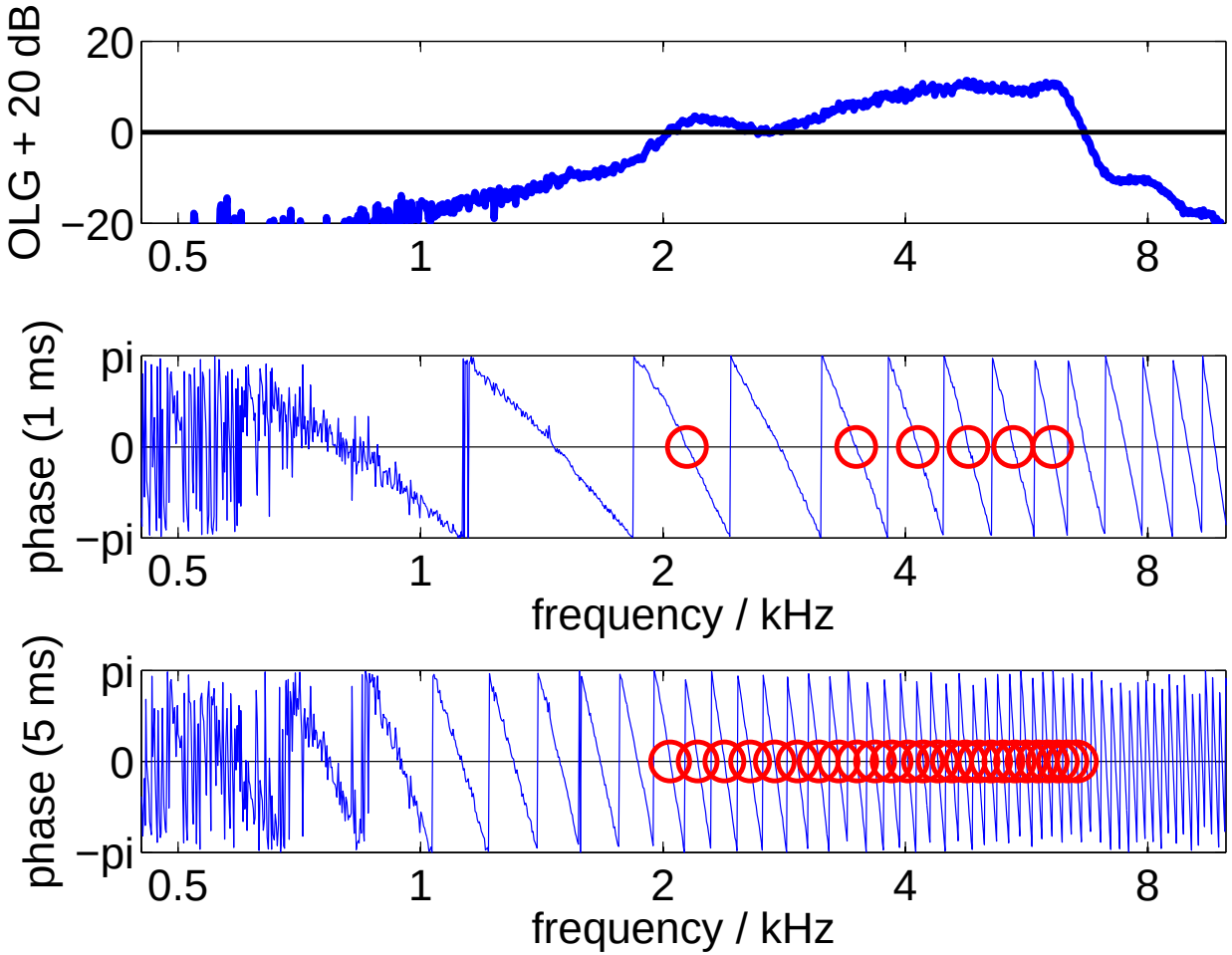
- Block processing delay:
  Fragment size plus processing delay
  (i.e., two blocks if input and output is block aligned)
- Hardware delay:
  Anti-aliasing filter (AD and DA, typical 1.5 ms in total)
  Data transmission, driver layer ('mystic' delays)
- Algorithmic delay:
  Group delay of filters, overlap-add delay, ...

Why do we aim for lowest delay? $\Rightarrow$ Feedback howling

- Feedback criterion: Roundtrip-gain is above 0 dB, and phase is multiple of 2π

Why do we aim for lowest delay? ⇒ Feedback howling

- Feedback criterion: Roundtrip-gain is above 0 dB, and phase is multiple of 2π



Achievable delays

Device	f_s/kHz	fragment size	τ_{sc}/ms	τ_t/ms
Echo Layla 3G	32	32	2.81	4.81
	44.1	64	2.04	4.94
	100	32	0.81	1.77
RME HDSP9652 + Behringer SRC2496	32	64	3.34	7.34
	44.1	64	2.68	5.58
	96	128	1.73	4.40
RME HDSP9652 + Behringer ADA8000	32	64	2.13	6.12
	44.1	64	1.61	4.51
RME HDSP9632 + ADI8QS	44.1	64	1.52	4.42
	96	128	1.03	3.7
OFFIS USB SC-4/2	16	16	6.81	9.81
	44.1	64	4.08	8.44
	96	128	1.97	5.97

Device	f_s/kHz	fragment size	τ_{sc}/ms	τ_t/ms
Echo Layla 3G	32	32	2.81	4.81
	44.1	64	2.04	4.94
	100	32	0.81	1.77
RME HDSP9652 + Behringer SRC2496	32	64	3.34	7.34
	44.1	64	2.68	5.58
	96	128	1.73	4.40
RME HDSP9652 + Behringer ADA8000	32	64	2.13	6.12
	44.1	64	1.61	4.51
RME HDSP9632 + ADI8QS	44.1	64	1.52	4.42
	96	128	1.03	3.7
OFFIS USB SC-4/2	16	16	6.81	9.81
	44.1	64	4.08	8.44
	96	128	1.97	5.97

Device	f_s/kHz	fragment size	τ_{sc}/ms	τ_t/ms
Echo Layla 3G	32	32	2.81	4.81
	44.1	64	2.04	4.94
	100	32	0.81	1.77
RME HDSP9652 + Behringer SRC2496	32	64	3.34	7.34
	44.1	64	2.68	5.58
	96	128	1.73	4.40
RME HDSP9652 + Behringer ADA8000	32	64	2.13	6.12
	48	64	1.46	4.12
RME HDSP9632 + ADI8QS	44.1	64	1.52	4.42
	96	128	1.03	3.7
OFFIS USB SC-4/2	16	16	6.81	9.81
	44.1	64	4.08	8.44
	96	128	1.97	5.97

Device	f_s/kHz	fragment size	τ_{sc}/ms	τ_t/ms
Echo Layla 3G	32	32	2.81	4.81
	44.1	64	2.04	4.94
	100	32	0.81	1.77
RME HDSP9652 +	32	64	3.34	7.34
Behringer SRC2496	44.1	64	2.68	5.58
	96	128	1.73	4.40
RME HDSP9652 +	32	64	2.13	6.12
Behringer ADA8000	44.1	64	1.61	4.51
RME HDSP9632 +	44.1	64	1.52	4.42
ADI8QS	96	128	1.03	3.7
OFFIS	16	16	6.81	9.81
USB SC-4/2	44.1	64	4.08	8.44
	96	128	1.97	5.97

Device	f_s/kHz	fragment size	τ_{sc}/ms	τ_t/ms
Echo Layla 3G	32	32	2.81	4.81
	44.1	64	2.04	4.94
	100	32	0.81	1.77
RME HDSP9652 +	32	64	3.34	7.34
Behringer SRC2496	44.1	64	2.68	5.58
	96	128	1.73	4.40
RME HDSP9652 +	32	64	2.13	6.12
Behringer ADA8000	44.1	64	1.61	4.51
RME HDSP9632 +	44.1	64	1.52	4.42
ADI8QS	96	128	1.03	3.7
OFFIS	16	16	6.81	9.81
USB SC-4/2	44.1	64	4.08	8.44
	96	128	1.97	5.97

Demonstration now!

Conclusions

- Hearing aid research is a wide field and requires careful evaluation.
- Linux Audio provides a valuable environment for hearing aid research.
- Low delay processing is possible with Linux/ALSA/Jack (but mystic delays remain in the chain of soft- and hardware).

Thank you for your attention!